

REMARKS

In an office action dated December 5, 2007, the Examiner objected to certain informalities in claims 5, 16, 22 and 26; and rejected claims 1-2, 4-5, 12-14, 16, 18 and 21-26 under 35 U.S.C. 102(e) as anticipated by McGuire et al. (U.S. Patent 6,493,871).

Applicants' Statement of Interview Substance

Applicants are in agreement with the Examiner's Interview Summary of March 17, 2008. As a matter of clarification, no agreement was reached as to specific allowable claim language (i.e., box "f" on form PTOL-413 appears to have been checked erroneously).

Claim Objections

Claims 5, 16 and 22 have been amended to correct the informalities noted by the Examiner. With respect to claim 26, applicants understand the objection to be that the claim is not in proper dependent form. Applicants have accordingly re-written claim 26 in what they believe to be correct form. If the Examiner has any further objection to the form of claim 26, applicants respectfully request further clarification as to the objectionable matter.

Prior Art

Applicants believe that the pre-existing claims sufficiently distinguish over *McGuire* and other art previously cited herein. However, in the interests of furthering prosecution herein, applicants have amended all independent claims to further define the scope of the present invention. These amendments are made without conceding that the existing claims are not patentable over *McGuire*, and applicant reserves the right to pursue these and other claims in one or more continuations or divisional patent applications, or to re-assert these claims in the present application in the event of further prosecution herein.

In particular, the claims are amended to recite that at least one upgrade, installation or maintenance object includes script instructions which instruct the user to manually perform an operation. Support for this limitation is found, inter alia, at Spec. p. 7, lines 8-16, p. 9, lines 7-19, and Fig. 3, block 308.

Applicants additionally note that independent claim 16 has been amended to recite “maintenance objects”, which may be broader in certain respects than the previously recited “upgrade objects”, and could perform any maintenance operation on the computer system. Dependent claim 29 recites specifically that the maintenance operation includes at least one of disk defragmentation, virus scanning, or data backup. Support for this limitation is found at Spec. p. 12, lines 14-17. No new matter is introduced.

As explained previously, applicants’ invention is intended to simplify computer system maintenance in a complex enterprise environment, particularly maintenance involving the installation and upgrading of computer software. Because a business or other enterprise may have many different systems and systems of different types, it is difficult to assure that installation, upgrade or other maintenance prerequisites will be met on all such systems, that configurations will be the same, and so forth. Enterprises often employ trained maintenance personnel for this purpose, but this is costly.

Applicants’ invention simplifies maintenance tasks by allowing a system administrator or similar person to create upgrade (installation/maintenance) objects in the form of scripts containing instructions is a pre-defined script instruction set. These instructions are not directly executable instructions, but require the use of a script processor for execution. However, they may contain conditional logic enabling operations of arbitrary complexity to be performed. The script processor is an executable program similar to a compiler or interpreter, which compiles and executes the script to perform the desired function. In particular, in accordance with the amended

independent claims, at least one of the operation performed by the executed script is to instruct or prompt the user to take some action manually (e.g. load a removable media, activate/deactivate a device, etc.). Thus, the script is not limited to actions which could be performed automatically by a computer processor, but can include a complex series of interactions between a user and computer system.

McGuire discloses a software upgrade technique designed to reduce the volume of data which must be downloaded to support an upgrade. In accordance with *McGuire*, the client to be upgraded initially downloads a “setup package”, which includes a setup program and a list of files (called a “script file”) required for the upgrade. The setup program executes on the client to determine which of the files listed in the “script file” is required to be downloaded. I.e., if a file is already present on the client, there is no need to download it. *McGuire* discloses that there could be other conditions on the downloading of a file. For example, there could be multiple script files corresponding to different upgrades, and a user could select which upgrades to perform, i.e. which script files to process. Furthermore, a script file may have an “always copy” section of files, which are always downloaded and copied, whether present or not, and a separate “replace if exist” section of files, which are downloaded and installed only if a version is already present.

Thus, it can be seen that in essence *McGuire* receives a list of files to be downloaded (which it calls a “script file”), and processes this list by downloading the listed files, with certain conditional logic one downloading a file. In this respect, *McGuire* is similar to the previously cited *Chiles* reference (which also disclosed an upgrade technique, in which an “update script” in the form of a list of files is downloaded from a server.

Applicants independent claims previously recited the use of the script having instructions of a pre-defined script instruction set which is executed by the script processing module, and further recited that the script contains at least one prerequisite allowing operations to be

performed out of sequence. The present amendment further adds the limitation that at least one operation performed by the script is an instruction to a user to perform a manual operation.

Applicants' representative claim 1, as amended, recites:

1. A method of upgrading a computer program installed on a first computer system, the computer program including a script processing module, the method comprising:

receiving a plurality of upgrade objects associated with the computer program, each said upgrade object including a respective script comprising a plurality of *script instructions of a pre-defined script instruction set* adapted for use by the script processing module to upgrade the computer program, the script being not independently executable without the script processing module, each said upgrade object being generated on a second computer system remote from said first computer system and transmitted from said second computer system to said first computer system, *wherein each said upgrade object contains a respective sequence number, wherein at least one said upgrade object contains at least one respective prerequisite, and wherein not all said upgrade objects have the same at least one respective prerequisite;*

with respect to each said upgrade object containing at least one respective prerequisite, automatically determining whether the at least one respective prerequisite has been met; and

performing an upgrade of said computer program by compiling and *executing each said script on said first computer system with the script processing module*, wherein, with respect to each said upgrade object containing at least one respective prerequisite, the respective script contained in the object is not compiled and executed until the respective at least one prerequisite contained in the object has been met, and *wherein the at least one prerequisite allows at least one script to be executed out of sequence;*

wherein at least one said upgrade object includes at least one script instruction of said pre-defined script instruction set which, when compiled and executed with said script processing module, *instructs a user to manually perform an operation for performing said upgrade.* [emphasis added]

Independent claim 12 contains analogous limitations to the italicized limitations above.

Independent claims 18 and 23 recite installing the program rather than upgrading it, the script processor being part of the environment to which the program is installed, but otherwise contain analogous recitations to the italicized limitations above. Independent claim 16 recites maintaining a computer system, which could involve upgrade, installation, or other maintenance operations, but otherwise contains analogous recitations to the italicized limitations above.

As explained above, *McGuire* discloses the use of a “script file”, which is essentially a list of files to be downloaded and optional simple conditions (“always download”, “download if present”, etc.) for downloading. *McGuire*’s setup program simply downloads all the files in the “always download” section, and downloads the files in the “download if present” if the corresponding file is present. Such a list of files does not amount to “instructions” of a “pre-defined script instruction set”, which are “executed” by a script processing module, as recited in applicants’ independent claims.

Furthermore, although *McGuire* does show some conditional aspect of downloading in that some files may be downloaded or skipped depending on whether they are already present, *McGuire* does not disclose verification of prerequisites and execution of operations out of sequence. The only conditionality shown is that a file is either downloaded or not, and this depends on a simple condition (file is already present or it isn’t). The claim limitation that a sequence number is assigned to upgrade objects, and they may be executed out of sequence depending on the satisfaction of specified prerequisites, is not shown in *McGuire*.

Finally, the newly added limitation that at least one script instructs the user to take an action is nowhere demonstrated in *McGuire*. This aspect provides significantly greater flexibility to perform complex upgrade or other maintenance operations using a combination of manually performed operations and computer-implemented functions, and allows the user to be guided to perform the operations in the correct sequence, even if the user has no particular expertise in the field. *McGuire* simply traverses a list of files to be downloaded, and automatically downloads (or skips) each listed file. All of the actions taken by *McGuire*’s setup program are intended to be performed automatically. In fact, the thrust of *McGuire* is to avoid relying on the user to perform operations.

For all of the reasons noted above, the independent claims are not anticipated by *McGuire*.

Nor are the independent claims obvious over *McGuire*. As explained, *McGuire*'s script amounts to multiple lists of files to download, each list corresponding to a different implied condition on download. The very limited conditional aspect of processing is a result of separate lists. While *McGuire* uses the word "script", the list of files does not amount to instructions of a pre-defined instruction set which are executed by a script processor, nor is there anything about a list of files which would suggest or otherwise render obvious such a "script" as used by applicants. Nor is there anything in *McGuire* which would suggest or otherwise render obvious instructing the user to perform some task as part of the upgrade/maintenance script, whereby a complex upgrade/maintenance operation can be performed as a combination of manual and machine executed operations.

In view of the foregoing, applicants submit that the claims are now in condition for allowance and respectfully request reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,
SAMUEL D. DULL III, et al.



By: _____
Roy W. Truelson
Registration No. 34,265

Telephone: (507) 289-6256 (Office/Fax)
(507) 202-8725 (Cell)